falls within the lower and upper threshold of bin 0. Thus the initial ECG value is encoded with the code 0. 12000 is transferred to next row as the serial accumulator for next ECG value. The next ECG value is 11904. The difference between the next ECG value and the serial accumulator for the second value is 11904−12000=−96. The difference of −96 falls into the bin with the code of −3, where the lower threshold of the bin is −41 and the upper threshold of the bin is −150. Thus, the second ECG value is encoded with the code of −3, which is the bin identification. For the purpose of decoding the second value, an encoder first refers to the assigned bin, which is bin −3; the encoder then reads the lower threshold ECG value of the assigned bin −3, which is −41; and the encoder finally add the lower threshold ECG value of the assigned bin to the decoded value of the first ECG value, which is 12000, to arrive at a decoded value of 11959. The decoded value 11959 in turn serves as the serial accumulator for the next ECG value, in this case the next ECG value is the third one of 12537. The difference between the third value and its corresponding serial accumulator is 12537−11959=578. This difference, 578, falls into the bin with a code of +5, which has a lower threshold ECG value of 301 and upper threshold ECG value of 1500. Thus the third ECG value is encoded with the code of +5. The third ECG value is decoded by adding the lower threshed ECG value of the assigned bin +5, which is 301, to the decoded value of second ECG value, which is 11959, to arrive at the decoded value of 12260. The decoded value of 12260 in turn will serve as the serial accumulator for the next ECG value. The encoding process continue until the last reading is taken. The encoder keeps track of the accumulated encoded value as the encoding process progresses along.

The encoding process described above is also a lossy compression process that encodes raw ECG signals with a finite number of codes. This process captures essential information while achieving significant data compression. In one embodiment, an other compressing step is performed. The other compression step may be performed independently. The other compression step may also be performed on top of the encoding process described above to achieve a higher level compression than one step alone. The second compression step can be a lossless compression performed on the codes from the first step. In one embodiment, the compression ratio of the second compression is in the range of 1.4 to 1.6, increasing the data storage capacity of a non-volatile memory by more than 41-66%. In another embodiment, the compression ratio of the second compression is in excess of 1.6, increasing the data storage capacity of a non-volatile memory by more than 66%. Thus, the combination of the lossy compression and the lossless compression serves to achieve both high fidelity of the ECG signal preservation and high compression ratio, which translate into increased data storage capacity and reduced power consumption for the ambulatory electrocardiography monitor, resulting in extended wear time of the monitor.

In one embodiment, the second compression is effected by encoding a sequence of codes obtained from the first compression into a single number between 0 and 1, with frequently used codes using fewer bits and not-so-frequently occurring codes using more bits, resulting in reduced storage space use in total. FIG. **26** is a flow diagram showing a monitor recorder-implemented method for further compressing the codes. A sequence of the codes corresponding to the series of the ECG values is provided to the compressing module **134**. The compressing module **134** set a range of 0 to 1 to an initial sequence of the codes (step **231**). The compressing module **134** further performs recursive steps of assigning each successive codes into a sub-range within a previous range according to the probabilities of the codes appearing after a code (steps **232**-**239**). In order to do so, the compressing module **134** obtains an estimation of probabilities of next codes, given a current code (step **233**). Several variations of calculating and adjusting the probabilities of the next codes will be described infra. The compressing module **134** divides the range of the current code into sub-ranges, each sub-range representing a fraction of the range proportional to the probabilities of the next codes (step **234**). These sub-ranges are contiguous and sequential. The compressing module **134** reads the next code (step **235**) and selects the sub-range corresponding to the read next code (step **236**). The read next code is represented, or encoded, by the corresponding sub-range (step **237**). The sub-range corresponding to the read next code is assigned to be the range for the code next to the read next code (step **238**), and the range is further divided into sub-ranges with each sub-range representing a fraction of the range proportional to the probabilities of codes next to the read next code (step **39**). In this way, each code in the sequence of the codes is represented by, or encoded through, its location within a sub-range through a recursive process. During the recursive process, strings of codes represented by the selected sub-ranges are encoded into part of the single number between 0 and 1 and can be periodically or continually stored into the non-volatile memory, can be stored on-demand or as-needed, or can be queued up and stored en masse upon completion of the process. One example of the non-volatile memory is the flash memory **62**.

The compressing module **134** uses a statistical model to predict what the next code is, given a current encoding (Step **233**). In one embodiment, a total of 16 codes or bin numbers are used, thus the statistical model uses 16 tables, one for each current code. Within each table, numeric possibilities for 16 possible next codes given the particular current code are generated. In one embodiment, the probabilities of the next codes can be calculated from sample ECG values. In another embodiment, the probabilities of the next codes can be modified by ECG data including recorded ECG data and data presently recorded. In still another embodiment, the probabilities of next codes can be adaptive, i.e, adjusted or varied along the recursive compression steps. Finally, in yet another embodiment, the compressing module **134** may use a statistical model to arrive at the estimation of probabilities of next codes, given two or more consecutive preceding codes. When two consecutive preceding codes are used, 16×16=256 different pairs of consecutive codes are possible. The compressing module **134** generates 256 tables, each tables containing numeric possibilities for 16 possible next codes given a particular pair of previous codes. When three consecutive preceding codes are used, 16×16×16=4096 different trios of consecutive codes are possible. The compressing module **134** generates 4096 tables, each tables containing numeric possibilities for 16 possible next codes given a particular trio of previous codes. Using two or more consecutive preceding codes further enhances compression ratio compared to using one preceding code, but also demands more processing power from the microcontroller.

While the invention has been particularly shown and described as referenced to the embodiments thereof, those skilled in the art will understand that the foregoing and other changes in form and detail may be made therein without departing from the spirit and scope.

What is claimed is:

1. A computer-implemented method for encoding and compressing electrocardiography values, comprising the steps of: